PROCESS FOR MANAGING DATA IN WHICH EXISTING DATA ITEM IS MOVED TO NEIGHBOR PAGE BEFORE INSERTION OR AFTER DELETION OF ANOTHER DATA ITEM

5   BACKGROUND OF THE INVENTION

1) Field of the Invention

The present invention relates to a process for managing a plurality of data items contained in a plurality of pages which are sequentially arranged in accordance with

10   predetermined relationships. The present invention also relates to a data management apparatus which executes the above process.

2) Description of the Related Art

15   In data processing systems such as database systems, in which data are written or from which data are deleted, data fragmentation in secondary storage devices increases with the passage of time, and access performance is lowered.

An example of fragmentation which occurs in a B-Tree

20   (balance tree) index (i.e., an index having a structure of a B-Tree) is explained below.

Fig. 10 is a diagram illustrating an example of a B-Tree index for a database. The fragmentation can occur in such an index. As illustrated in Fig. 10, a B-Tree index is

25   comprised of a root page at the top level, index pages at an intermediate level or levels, and leaf pages at the bottom level. That is, the B-Tree index has a tree

structure in which a plurality of branches extend from each
node to a plurality of descendants of the node at lower
levels, and a page is arranged at each node. Information
indicating a range of values of keys is stored in each of
5   the root page and the index pages so as to reach a leaf
page based on the value of a key. In addition, at least one
index data item each of which locates a record
corresponding to a key are stored in each leaf page.

In the root page, the range from 0 to 50 is
10 represented by "50", and the range from 51 to 100 is
represented by "100". Therefore, when the value of a key is
"10", the operation goes to the left index page. Since, in
the left index page, the range from 0 to 30 is represented
by "30", and the range from 31 to 50 is represented by "50",
15 the operation then goes to the leftmost leaf page, which
contains index data items "10" and "30". Thus, a record
stored in a secondary storage device and located by the
index data item "10" can be accessed.

When a new data item is added to a database which has
20 a B-Tree index, or a data item is deleted from such a
database, the B-Tree index per se must be modified
accordingly. Examples of conventional methods of
modification of a B-Tree index corresponding to addition of
a data item to a database or deletion of a data item from a
25 database are explained below.

Figs. 11(A) through 11(C) are diagrams illustrating
an example of a sequence of operations performed for

- 2 -

modifying leaf pages in a B-Tree index for a database when a new data item is added to the database. When an index data item 20 having a value "60" is inserted into the leaf page 11 among the sequentially arranged leaf pages 10 to 12 as illustrated in Fig. 11(A), it is necessary to create a new leaf page 13 since there is no available space in the leaf page 11. Next, in order to obtain space for storing the new dynamic image data 20, two index data items respectively having values "80" and "90" in the leaf page 11 are moved to the newly created leaf page 13, as illustrated in Fig. 11(B). Then, the new index data item 20 is written in the obtained space in the leaf page 11, as illustrated in Fig. 11(C).

Figs. 12(A) and 12(B) are diagrams illustrating an example of a sequence of operations performed for modifying leaf pages in a B-Tree index for a database when an existing data item is deleted from the database. When an index data item having a value "50" is deleted from the leaf page 11 among the sequentially arranged leaf pages 10 to 12 as illustrated in Fig. 12(A), no index data item remains in the leaf page 11 as illustrated in Fig. 12(B). Therefore, the leaf page 11 becomes a degenerated page, and is returned to the system.

As explained above, when a new index data item is inserted into a B-Tree index, often, fragmentation occurs in the leaf pages, i.e., short runs of space are generated in the leaf pages. Each leaf page having a short run of

space remains to exist until the space is filled with a new index data item, or an index data item or index data items existing in the leaf page are deleted as illustrated in Figs. 12(A) and 12(B), and the leaf page is returned as a

5 degenerated page to the system. Further, fragmentation also occurs in the index pages.

Conventionally, when fragmentation occurs, the operation of the system is stopped, and the data items are rearranged by executing an defragmentation program. However,

10 the frequency of occurrence of fragmentation varies with the circumstances. Therefore, a system management person is required to monitor occurrence of fragmentation.

In addition, many database systems and the like which are currently used are operated in a nonstop mode.

15 Therefore, it is difficult to stop the operation of the system for performing defragmentation.

SUMMARY OF THE INVENTION

The object of the present invention is to provide a

20 process for managing a plurality of data items contained in a plurality of pages which are sequentially arranged in accordance with predetermined relationships, whereby defragmentation is automatically performed without stopping the system, and a load imposed on a system management

25 person is reduced.

(1) According to the first aspect of the present invention, there is provided a process for managing a

plurality of data items contained in a plurality of pages which are sequentially arranged in accordance with predetermined relationships, comprising the steps of: (a) storing information on an amount of available space in each of the plurality of pages; (b) acquiring an amount of available space in each of first and second pages based on the information stored in step (a), when a first data item is inserted into or deleted from a third page, the plurality of pages include the first, second, and third pages, the first page precedes the third page in the sequential arrangement of the plurality of pages, and the second page follows the third page in the sequential arrangement; and (c) moving at least one second data item contained in the third page to the first page or the second page according to the amount of available space in each of the first and second pages, before insertion of the first data item into the third page or after deletion of the first data item from the third page.

The process according to the first aspect of the present invention may have one or any possible combination of the following additional features (i) to (iv).

(i) In the step (a), when the amount of the available space is equal to or less than a predetermined amount, the information may indicate that substantially no available space exists.

(ii) In the step (c), when both of the first and second pages have sufficient available space, the at

least one second data item may be moved to the second page.

(iii) In the step (a), the amount of the available space may be classified into one of a plurality of ranges of amounts of the available space, and the information on the amount of the available space may indicate one of the plurality of ranges.

(iv) One of the plurality of ranges including the smallest amount of the available space may be wider than the other of the plurality of ranges.

(2) According to the second aspect of the present invention, there is provided a data management apparatus for managing a plurality of data items contained in a plurality of pages which are sequentially arranged in accordance with predetermined relationships, comprising: an available-space-information storing unit which stores information on an amount of available space in each of the plurality of pages; an adjacent-space-information acquiring unit which acquires an amount of available space in each of first and second pages based on the information stored in the available-space-information storing unit, when a first data item is inserted into or deleted from a third page, the plurality of pages include the first, second, and third pages, the first page precedes the third page in the sequential arrangement of the plurality of pages, and the second page follows the third page in the sequential arrangement; and a moving unit which moves at least one second data item contained in the third page to the first

page or the second page according to the amount of available space in each of the first and second pages, before insertion of the first data item into the third page or after deletion of the first data item from the third

5   page.

The data management apparatus according to the second aspect of the present invention may have one or any possible combination of the aforementioned additional features (i) to (iv).

10   (3) According to the first and second aspects of the present invention, at least one data item located around a deleted data item or a data item which is to be inserted can be rearranged at the time of the insertion or deletion. Therefore, fragmentation can be suppressed without stopping

15   the system, and access performance can be improved. In addition, defragmentation can be efficiently performed, since the rearrangement is performed on only the at least one data item located around the deleted data item or the data item to be inserted according to the present invention,

20   while the conventional defragmentation is performed on the entire storage area.

(4) According to the third aspect of the present invention, there is provided a computer-readable storage medium storing a program which makes a computer execute a

25   process for managing a plurality of data items contained in a plurality of pages which are sequentially arranged in accordance with predetermined relationships, the program

further makes the computer realize: an available-space-information storing unit which stores information on an amount of available space in each of the plurality of pages; an adjacent-space-information acquiring unit which acquires an amount of available space in each of first and second pages based on the information stored in the available-space-information storing unit, when a first data item is inserted into or deleted from a third page, the plurality of pages include the first, second, and third pages, the first page precedes the third page in the sequential arrangement of the plurality of pages, and the second page follows the third page in the sequential arrangement; and a moving unit which moves at least one second data item contained in the third page to the first page or the second page according to the amount of available space in each of the first and second pages, before insertion of the first data item into the third page or after deletion of the first data item from the third page.

The computer-readable storage medium according to the third aspect of the present invention may have one or any possible combination of the aforementioned additional features (i) to (iv).

The above and other objects, features and advantages of the present invention will become apparent from the following description when taken in conjunction with the accompanying drawings which illustrate preferred embodiment

of the present invention by way of example.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings:

5   Fig. 1 is a diagram illustrating the principle of the present invention;

Fig. 2 is a diagram illustrating an example of a hardware construction of the data management apparatus according to the present invention;

10   Fig. 3(A) is a diagram illustrating an example of an available-space management table;

Fig. 3(B) is a diagram illustrating examples of indications of page occupation ratios in the available-space management table of Fig. 3(A);

15   Fig. 4(A) is a diagram illustrating examples of sequentially arranged leaf pages before insertion of an index data item into a leaf page;

Fig. 4(B) is a diagram illustrating examples of sequentially arranged leaf pages after the insertion of the

20   index data item into the leaf page;

Fig. 5(A) is a diagram illustrating an example of an available-space management table before the insertion of the index data item explained with reference to Figs. 4(A) and 4(B);

25   Fig. 5(B) is a diagram illustrating an example of the available-space management table after the insertion of the index data item explained with reference to Figs. 4(A) and

4(B);

Fig. 6(A) is a diagram illustrating examples of sequentially arranged leaf pages before deletion of an index data item from a leaf page;

5 Fig. 6(B) is a diagram illustrating examples of sequentially arranged leaf pages after the deletion of the index data item from the leaf page;

Fig. 7(A) is a diagram illustrating an example of an available-space management table before the deletion of the

10 index data item explained with reference to Figs. 6(A) and 6(B);

Fig. 7(B) is a diagram illustrating an example of the available-space management table after the deletion of the index data item explained with reference to Figs. 6(A) and

15 6(B);

Fig. 8 is a flow diagram illustrating a sequence of operations for inserting an index data item into a (leaf) page of an index for a database when a new data item is added to the database;

20 Fig. 9 is a flow diagram illustrating a sequence of operations for deleting an index data item from a (leaf) page of an index for a database when a data item is deleted from the database;

Fig. 10 is a diagram illustrating an example of a B-

25 Tree index for a database;

Figs. 11(A) to 11(C) are diagrams illustrating an example of a sequence of operations performed for modifying

leaf pages in a B-Tree index for a database when a new data item is added to the database; and

Figs. 12(A) and 12(B) are diagrams illustrating an example of a sequence of operations performed for modifying leaf pages in a B-Tree index for a database when an existing data item is deleted from the database.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embodiments of the present invention are explained in detail below with reference to drawings.

(1) Functions of Data Management Apparatuses

Fig. 1 is a diagram illustrating the principle of the present invention. In Fig. 1, reference numeral 1 denotes a data management apparatus according to the present invention, and 2 denotes a database which stores a plurality of data items in a plurality of pages (including the pages 2a to 2c illustrated in Fig. 1) sequentially arranged in accordance with predetermined relationships. The data management apparatus 1 comprises an available-space-information storing unit 1a, an available-space acquiring unit 1b, and a movement unit 1c, and manages the data stored in the database 2.

The available-space-information storing unit 1a stores information on an amount of available space in each page, where each page is a storage area in which a predetermined amount of data can be stored. When a first data item is inserted into or deleted from a first page

among the plurality of pages, the available-space acquiring unit 1b acquires an amount of available space in each of second and third pages among the plurality of pages by referring to the available-space-information storing unit 1a, where the second page precedes the first page in the sequential arrangement of the plurality of pages, and the third page follows the second page in the sequential arrangement. The movement unit 1c moves at least one second data item contained in the first page to the second page or the third page according to the amount of available space in each of the second and third pages, before insertion of the first data item into the first page or after deletion of the first data item from the first page.

The operations of the data management apparatus 1 according to the present invention which are performed when a new (first) data item is inserted into the page 2b are explained below.

First, the available-space acquiring unit 1b refers to the available-space-information storing unit 1a, and acquires the amount of available space in the page 2b into which the first data item is to be inserted. In the example of Fig. 1, the page occupation ratio of the page 2b is 100% (as indicated in the upper row of the pages 2a to 2c in Fig. 1), i.e., there is no available space in the page 2b. When the available-space acquiring unit 1b recognizes that there is no available space in the page 2b, the available-space acquiring unit 1b refers to the available-space-information

storing unit 1a, and acquires the amount of available space in each of the page 2a which precedes the page 2b and the page 2c which follows the page 2b. As indicated in the upper row of the pages 2a to 2c in Fig. 1, the page occupation ratio of the page 2a is 80%, i.e., the amount of available space in the page 2c is 20%, and the page occupation ratio of the page 2c is 50%, i.e., the amount of available space in the page 2c is 50%.

Next, the movement unit 1c determines to which of the pages 2a and 2c a portion or all (at least one second data item) of the data existing in the page 2b is to be moved, based on the data amount of the first data item to be inserted into the page 2b and the amount of available space in each of the pages 2a and 2c acquired by the available-space acquiring unit 1b. In this example, when the data amount of the first data item corresponds to 30% of the capacity of the page 2b, the movement unit 1c determines that a portion of the data existing in the page 2b is to be moved to the page 2c, and moves the portion to the page 2c, where the amount of the moved portion corresponds to 30% of the capacity of the page 2b. Accordingly, available space corresponding to 30% of the capacity of the page 2b is obtained in the page 2b, and thus the first data item can be inserted into the page 2b.

As explained above, according to the present invention, when there is no available space in a target page (i.e., a page into which a new data item is to be

- 13 -

inserted), a necessary amount of data in the target page is moved to a neighbor page. Therefore, the new data item can be inserted into the target page without creating a new page. Thus, increase in fragmentation can be prevented.

(2) Hardware Construction

Fig. 2 is a diagram illustrating an example of a hardware construction of the data management apparatus as an embodiment of the present invention. In Fig. 2, reference numeral 30 denotes a server as the data management apparatus, 31, 32, and 33 each denote a database (DB), 34 denotes a network, and 35, 36, and 37 each denote a client.

The server 30 comprises a central processing unit (CPU) 30a, a read only memory (ROM) 30b, a random access memory (RAM) 30c, a hard disk drive (HDD) 30d, a bus 30e, and interfaces (I/F) 30f and 30g. The server 30 manages the databases 31 to 33. In response to a request from one of the clients 35 to 37, the server 30 writes data in the databases 31 to 33, and reads data from the databases 31 to 33.

The CPU 30a controls the respective portions of the server 30, and executes various kinds of processing in accordance with programs and data stored in HDD 30d and the like. The ROM 30b stores basic programs to be executed by the CPU 30a and data used in the execution of programs. The RAM 30c temporarily stores a currently executed program and

data relating to the execution. The HDD 30d stores data and programs to be executed by the CPU 30a. The bus 30e interconnects the CPU 30a, the ROM 30b, the RAM 30c, the HDD 30d, and the interfaces 30f and 30g so as to enable

5　data transfer between the CPU 30a, the ROM 30b, the RAM 30c, the HDD 30d, and the interfaces 30f and 30g. The interface 30f electrically connects the server 30 with the databases 31 to 33 so that data can be transferred between the server 30 and the databases 31 to 33. When data are exchanged

10　between the server 30 and the network 34, the interface 30g transforms data forms, and converts protocols. The network 34 is, for example, the Internet, and data is transmitted through the network 34 between the server 30 and the clients 35 to 37. The clients 35 to 37 are realized by, for

15　example, personal computers, and access the server 30 in order to write data in and read data from the databases 31 to 33.

The databases 31 to 33 are constituted by Hard Disk Drives or the like, and store data and B-Tree indexes for

20　use in management of the data.

(3) Page Occupation Rate

A page number is assigned to each of pages (such as the index pages and leaf pages) constituting the B-Tree

25　index for each of the databases 31 to 33. Directories of the databases 31 to 33 contain available-space management tables, each of which indicates an occupation rate of each

page of the B-Tree index. For example, in each available-space management table, the page is indicated by the page number, and the occupation rate is indicated by two-bit data, as illustrated in Fig. 3(A). The two-bit data indicates a range of the occupation rates as illustrated in Fig. 3(B), where the occupation rate is expressed as percent. That is, in this example, the occupation rate in the range of 0 to 19% is represented by the two-bit data "11", the occupation rate in the range of 20 to 39% is represented by the two-bit data "10", the occupation rate in the range of 40 to 59% is represented by the two-bit data "01", and the occupation rate in the range of 60 to 100% is represented by the two-bit data "00". In order to prevent frequent occurrence of the movement of the index data item, consolidation of two leaf pages, and the like, the range including the highest occupation rates 60 to 100% is wider than the other ranges.

(4) Insertion of Index Data Item into Leaf Page

An operation of inserting an index data item into a leaf page (hereinbelow called a page) in a B-Tree index is explained below with reference to Figs. 4(A), 4(B), 5(A), and 5(B).

When an instruction to add a new data item to the database 31 is sent from one of the clients 35 to 37 to the server 30, the CPU 30a in the server 30 receives the instruction, and determines a target page, i.e., a page

into which a new index data item corresponding to the new data item is to be inserted, by referring to the B-Tree index of the database 31. For example, when the page 51 illustrated in Fig. 4(A) is determined to be the target page, the CPU 30a refers to the available-space management table (as explained with reference to Fig. 3(A)), and acquires information on the occupation rate of the page 51. Fig. 5(A) is a diagram illustrating an example of the available-space management table before the insertion of the index data item. Since, in this example, there is no available space in the page 51 as illustrated in Fig. 4(A), the available-space management table indicates "00" for the page 51. When the CPU 30a recognizes that there is no available space in the page 51, the CPU 30a acquires information on the occupation rates of the pages 50 and 52 by referring to the available-space management table, where the page 50 precedes the page 51, and the page 52 follows the page 51. Since, in this example, the available space in the page 52 amounts to 70% of the full capacity as illustrated in Fig. 4(A), the available-space management table indicates "10" for the page 52 as illustrated in Fig. 5(A). Therefore, the CPU 30a moves a portion of data in the page 51 to the page 52 so as to obtain necessary space in the page 51. That is, the CPU 30a calculates a data amount required to be moved for insertion of the new index data item into the page 51, and moves to the page 52 a portion or all of data (at least one index data item) in the page

51 corresponding to the calculated data amount. For example, when the amount of the new index data item to be inserted into the page 51 is calculated to be 40%, a portion of data (at least one index data item) in the page 51 corresponding to 40% of the full capacity of the page 51 is moved to the page 52. As a result, available space corresponding to 40% of the full capacity is obtained in the page 51 as illustrated in Fig. 4(B). Then, the CPU 30a inserts the new index data item into the page 51, and rewrites the indications (two-bit data) of the occupation rates of the pages 51 and 52 in the available-space management table as illustrated in Fig. 5(B). Thereafter, the contents of the index pages in the B-Tree index are updated corresponding to the changes in the contents of the leaf pages.

As explained above, according to the present invention, when a new index data item is requested to be inserted into a target page, the amount of available space in the target page is detected. Then, when there is not sufficient available space in the target page, it is determined whether or not sufficient available space exists in either of the preceding and following pages. When sufficient available space exits in one of the preceding and following pages, a portion or all of the data existing in the target page are moved to the one of the preceding and following pages, and then the new index data item is inserted into the target page. Therefore, when sufficient available space exits in at least one of the preceding and

- 18 -

following pages, it is not necessary to create a new page, and fragmentation can be reduced to a minimum.

In addition, since defragmentation is automatically performed when a new index data item is inserted into a page, it is not necessary to stop the operation of the system for defragmentation. Therefore, the system management person can be relieved from the work of monitoring the condition of the system.

(5) Deletion of Index Data Item from Leaf Page

An operation of deleting an index data item from a leaf page (hereinbelow called a page) in a B-Tree index is explained below with reference to Figs. 6(A), 6(B), 7(A), and 7(B).

When an instruction to delete a data item from the database 31 is sent from one of the clients 35 to 37 to the server 30, the CPU 30a in the server 30 receives the instruction, and determines a target page, i.e., a page from which an index data item corresponding to the data item to be deleted from the database 31 is to be deleted, by referring to the B-Tree index of the database 31. For example, when the page 51 illustrated in Fig. 6(A) is determined to be the target page, the CPU 30a deletes the index data item from the target page, and updates the indication of the occupation rate of the page 51 in the available-space management table. Since, in this example, available space corresponding to 50% of the full capacity

remains in the page 51 as illustrated in Fig. 6(A), the indication of the occupation rate of the page 51 is updated to "01", as illustrated in Fig. 7(A). Next, the CPU 30a acquires information on the occupation rates of the pages 5 50 preceding the page 51 and the page 52 following the page 51, by referring to the available-space management table, and determines whether or not all of the data (at least one index data item) remaining in the target page 51 can be moved to the preceding page 50 or the following page 52. 10 Since, in this example, available space in the page 52 is 70% of the full capacity, the CPU 30a determines that all of the data remaining in the target page 51 can be moved to the page 52, and the movement is made. Accordingly, no data (index data item) remains in the page 51 as illustrated in 15 Fig. 6(B). Therefore, the CPU 30a returns to the system the page 51 as a degenerated page, and updates the available-space management table as illustrated in Fig. 7(B). Thereafter, the contents of the index pages in the B-Tree index are updated corresponding to the changes in the 20 contents of the leaf pages.

As explained above, according to the present invention, when an index data item is requested to be deleted from a target page, the amount of remaining data in the target page after the deletion and the amount of 25 available space in each of the preceding and following pages are detected. When it is determined that all of the data remaining in the target page can be moved to the

preceding page or the following page, the movement is made, and the vacated page is returned to the system. Therefore, fragmentation can be reduced to a minimum.

In addition, since defragmentation is automatically performed when an index data item is deleted from a page, it is not necessary to stop the operation of the system for defragmentation. Therefore, the system management person can be relieved from the work of monitoring the condition of the system.

(6) Sequence of Insertion of Index Data Item into Leaf Page

Fig. 8 is a flow diagram illustrating a sequence of operations for inserting an index data item into a (leaf) page of an index for a database when a new data item is added to the database.

In step S10, the CPU 30a determines a target page (leaf page) into which a new index data item corresponding to the data item to be added to the database is to be inserted, by referring to the B-Tree index for the database. In step S11, the CPU 30a acquires information on the occupation rate of the target page, by referring to the available-space management table. In step S12, the CPU 30a determines, based on the acquired information on the occupation rate of the target page, whether or not the target page includes available space which is sufficient for the insertion of the new index data item. When yes is determined in step S12, the operation goes to step S18.

When no is determined in step S12, the operation goes to step S13. In step S13, the CPU 30a acquires information on occupation rates of a preceding page and a following page from the available-space management table, where, in the sequential arrangement of the plurality of leaf pages of the index, the preceding page precedes the target page, and the following page follows the target page. In step S14, the CPU 30a determines, based on the acquired information on the occupation rates of the preceding page and the following page, whether or not one of the preceding page and the following page includes available space which is sufficient for receiving a portion or all (at least one index data item) of the data currently stored in the target page, where the amount of the portion or all of the data is determined so that the new index data item can be inserted into the target page after the portion or all of the data are moved from the target page to one of the preceding page and the following page. When yes is determined in step S14, the operation goes to step S17. When no is determined in step S14, the operation goes to step S15. In step S15, the CPU 30a creates a new page (leaf page), and inserts the new page immediately after the target page. In step S16, the CPU 30a moves to the new page a portion or all of the data currently stored in the target page, where the amount of the portion or all of the data is determined so that the new index data item can be inserted into the target page after the portion or all of the data are moved from the

- 22 -

target page to the new page. Then, the CPU 30a inserts the new index data item into the target page. In step S17, the CPU 30a moves a portion or all of the data currently stored in the target page to one of the preceding page and the following page, where the amount of the portion or all of the data is determined so that the new index data item can be inserted into the target page after the portion or all of the data are moved from the target page to the one of the preceding page and the following page. When both of preceding page and the following page have sufficient available space, the above portion or all of the data in the target page are moved to the following page in consideration of cost. In step S18, the CPU 30a inserts the new index data item into the target page. In step S19, the CPU 30a updates the available-space management table corresponding to the above changes in the (leaf) pages.

(7) Sequence of Deletion of Index Data Item from Leaf Page

Fig. 9 is a flow diagram illustrating a sequence of operations for deleting an index data item from a (leaf) page of an index for a database when a data item is deleted from the database.

In step S30, the CPU 30a determines a target page from which an index data item corresponding to the data item to be deleted from the database 31 is to be deleted, by referring to the B-Tree index for the database. In step S31, the CPU 30a deletes the index data item from the

target page. In step S32, the CPU 30a updates the contents of the available-space management table corresponding to the deletion of the index data item from the target page. In step S33, the CPU 30a acquires information on occupation rates of a preceding page and a following page from the available-space management table, where, in the sequential arrangement of the plurality of leaf pages of the index, the preceding page precedes the target page, and the following page follows the target page. In step S34, the CPU 30a determines whether or not all of the data (at least one index data item) remaining in the target page can be moved to the preceding page or the following page. When yes is determined in step S34, the operation goes to step S35. When no is determined in step S34, the operation of Fig. 9 is completed. In step S35, the CPU 30a moves all of the data remaining in the target page to one of the preceding page and the following page which is determined in step S34. When it is determined that all of the data remaining in the target page can be moved to each of the preceding page and the following page, the data are moved to the following page in consideration of cost. In step S36, the CPU 30a returns the vacated target page to the system, as a degenerated page which can be reused. In step S37, the CPU 30a updates the available-space management table corresponding to the above changes in the (leaf) pages.

According to the sequences of Figs. 8 and 9, fragmentation can be reduced during the operations of

insertion or deletion of data.

(8) Advantages of Embodiment

According to the present invention, when a new data item is added to a database, at least one index data item in a target page of an index is moved to one of a preceding page or a following page, as necessary, before a new index data item corresponding to the new data item is inserted into the target page. In addition, when a data item is deleted from a database, all of at least one index data item remaining in a target page of an index after deletion of an index data item corresponding to the deleted data item from the target page is moved to one of a preceding page or a following page, where possible. Therefore, fragmentation in the storage area of the index can be reduced to a minimum. As a result, it is possible to minimize the storage area necessary to store the index since the number of pages of the index is minimized. In addition, the access performance in a sequential search of pages in the lateral direction can be improved when the sequential search is performed based on a predetermined range of values of a key.

(9) Recording Medium Storing Program

(i) In the above embodiment, the present invention is applied to processing for insertion and deletion of an index data item into a page in a B-Tree index. However, the

present invention can be applied to processing for insertion and deletion of data in any system which handles a plurality of data items contained in a plurality of pages which are sequentially arranged in accordance with predetermined relationships.

(ii) The functions of the data management apparatus according to the present invention can be realized by a computer. The functions of the data management apparatus can be written as a program, which can be stored in a computer-readable storage medium. The functions can be realized by a computer by executing the program. The computer-readable storage medium may be a magnetic storage device, a semiconductor memory, or the like. In order to put the program into the market, the program may be stored in a portable storage medium such as a CD-ROM (compact-disk read-only memory) and a floppy disc. Alternatively, the programs can be stored in a storage device belonging to a first computer, and transferred to a second computer connected through a network to the first computer. When the program is executed by a computer, the program can be stored in a hard disk drive or the like belonging to the computer, and loaded into a main memory of the computer.

(iii) The foregoing is considered as illustrative only of the principle of the present invention. Further, since numerous modifications and changes will readily occur to those skilled in the art, it is not desired to limit the invention to the exact construction and applications shown

and described, and accordingly, all suitable modifications and equivalents may be regarded as falling within the scope of the invention in the appended claims and their equivalents.

5   (iv) In addition, all of the contents of the Japanese patent application, No.2000-301676 are incorporated into this specification by reference.